CLAIMS

1. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a cluster engine residing in the memory and executed by the at least one processor;

a job residing in the memory and executed by the at least one processor, the job including:

at least one work thread that performs at least one predefined task; and

a main thread that receives messages from at least one computer system coupled to the apparatus, that routes appropriate messages from the at least one computer system to the at least one work thread, and that signals to the cluster engine when at least one fault occurs when the at least one work thread performs the at least one predefined task.

2. The apparatus of claim 1 wherein the at least one predefined task comprises a protocol that includes at least one acknowledge (ACK) round, and that performs only local processing between ACK rounds.

3. The apparatus of claim 1 wherein the main thread performs only local processing.

4. The apparatus of claim 1 wherein the main thread does not wait for any local resource, and thus is guaranteed to receive a message sent by the cluster engine.

5. The apparatus of claim 1 wherein the signal to the cluster engine comprises an unregistration with the cluster engine

6. The apparatus of claim 5 wherein the unregistration with the cluster engine causes the cluster engine to generate a membership change message.

7. A networked computer system comprising:

a cluster of computer systems that each includes:

a network interface that couples each computer system via a network to other computer systems in the cluster;

a memory;

a cluster engine residing in the memory and executed by the at least one processor;

a job residing in the memory and executed by the at least one processor, the job including:

at least one work thread that executes a predefined protocol that includes at least one acknowledge (ACK) round, wherein the protocol only performs local tasks between ACK rounds; and

a main thread that registers with the cluster engine to become a member of a group, that receives messages from at least one computer system coupled to the apparatus, that routes appropriate messages from the at least one computer system to the at least one work thread, and that signals to the cluster engine when at least one fault occurs when the at least one work thread performs the at least one predefined task by unregistering with the cluster engine, wherein unregistering with the cluster engine causes the cluster engine to generate a membership change to remaining members of the group.

8. A computer-implemented method for notifying jobs that form a group in a clustered computing environment when a member of the group is no longer alive, the method comprising the steps of:

defining a task;

providing a cluster engine for each member of the group that communicates with the other cluster engines in the group;

providing at least one work thread for each job that executes the task;

providing a main thread for each job, the main thread performing the steps of:

receiving messages from other members of the group via the cluster engine corresponding to the main thread;

routing appropriate messages from the other members of the group to the at least one work thread; and

signaling to the cluster engine when at least one fault occurs during the execution of the task by the work thread.

9. The method of claim 8 wherein the task comprises a protocol that includes at least one acknowledge (ACK) round, and that performs only local processing between ACK rounds.

10. The method of claim 8 wherein the main thread performs only local processing.

11. The method of claim 8 wherein the main thread does not wait for any local resource, and thus is guaranteed to receive a message sent by the cluster engine.

12. The method of claim 8 wherein the step of signaling to the cluster engine comprises the step of unregistering with the cluster engine.

13. The method of claim 12 wherein the step of unregistering with the cluster engine causes the cluster engine to generate a membership change message to remaining members of the group.

14. A computer-implemented method for notifying jobs that form a group in a clustered computing environment when a member of the group is no longer alive, the method comprising the steps of:

defining a protocol that includes at least one acknowledge (ACK) round, and that performs only local processing between ACK rounds;

providing a cluster engine for each member of the group that communicates with the other cluster engines in the group;

providing at least one work thread for each job that executes at least a portion of the protocol;

providing a main thread for each job, the main thread performing the steps of:

registering with the cluster engine to become a member of the group;

receiving messages from other members of the group via the cluster engine corresponding to the main thread;

routing appropriate messages from the other members of the group to the at least one work thread;

wherein the main thread performs only local processing and does not wait for any local resource, and thus is guaranteed to receive a message sent by the cluster engine;

unregistering with the cluster engine when at least one fault occurs during execution of the protocol.

15. The method of claim 14 wherein the step of unregistering with the cluster engine causes the cluster engine to generate a membership change message to remaining members of the group.

16. A program product comprising:

(A) a computer program comprising:

at least one work thread that performs at least one predefined task; and

a main thread that receives messages from a corresponding cluster engine, that routes appropriate messages from the cluster engine to the at least one work thread, and that signals to the cluster engine when at least one fault occurs when the at least one work thread performs the at least one predefined task; and

(B) signal bearing media bearing the computer program.

17. The program product of claim 16 wherein the signal bearing media comprises recordable media.

18. The program product of claim 16 wherein the signal bearing media comprises transmission media.

19. The program product of claim 16 wherein the at least one predefined task comprises a protocol that includes at least one acknowledge (ACK) round, and that performs only local processing between ACK rounds.

20. The program product of claim 16 wherein the main thread performs only local processing.

21. The program product of claim 16 wherein the main thread does not wait for any local resource, and thus is guaranteed to receive a message sent by the cluster engine.

22. The program product of claim 16 wherein the signal to the cluster engine comprises an unregistration with the cluster engine

23. The program product of claim 22 wherein the unregistration with the cluster engine causes the cluster engine to generate a membership change message.

24. A program product comprising:

(A) a computer program comprising:

at least one work thread that performs a predefined protocol that includes at least one acknowledge (ACK) round, wherein the protocol only performs local tasks between ACK rounds; and

a main thread that registers with the cluster engine to become a member of a group, that receives messages from at least one computer system coupled to the apparatus, that routes appropriate messages from the at least one computer system to the at least one work thread, and that signals to the cluster engine when at least one fault occurs when the at least one work thread performs the at least one predefined task by unregistering with the cluster engine, wherein unregistering with the cluster engine causes the cluster engine to generate a membership change to remaining members of the group; and

(B) signal bearing media bearing the computer program.

25. The program product of claim 24 wherein the signal bearing media comprises recordable media.

26. The program product of claim 24 wherein the signal bearing media comprises transmission media.

*****